ABSTRACT

To alleviate the problems associated with modifying switching software for each individual hardware components, a logical switch abstraction is provide that is separated from an underlying physical switch abstraction, the physical abstraction being dependent upon the underlying components used in the switch. The abstraction is a model of the connection paths and switching elements of the switch. By efficiently determining connections within the logical abstraction and mapping those connections in the physical abstraction, changes in underlying hardware has a minimal effect on switching software. That is, adding new hardware to the switch has minimal effect on how connections are determined through the logical abstraction. More particularly, when a hardware type is changed or added, only mapping information identifying relations between components in the logical and physical abstractions changes. Because the logical abstraction is independent of the hardware implementation, connections are more easily managed. Further, an efficient method of provisioning is provided wherein the amount of connection time is reduced.